Nota: Los ejercicios que tienen (S) son para una "Segunda vuelta" es decir conviene hacerlos una vez que ya se completó la guía haciendo los otros y ya se tiene mas madurez e intuición basica sobre los conceptos. Los que tienen (O) son opcionales por lo cual no se toman en los examenes.

Tres paradigmas matematicos de la computabilidad efectiva

Ya que el concepto de procedimiento efectivo es un concepto intuitivo, impresiso y a priori no expresado en el formalismo matematico, los conceptos de

- Funcion Σ-efectivamente computable
- Conjunto Σ-efectivamente computable
- Conjunto Σ-efectivamente enumerable

tambien son impresisos y estan fuera del formalismo matematico, debido a que los tres se definen en terminos de la existencia de procedimientos efectivos. Por supuesto, los tres conceptos son fundamentales en el estudio teorico de la computabilidad por lo que es muy importante poder dar un modelo o formalizacion matematica de estos conceptos. Pero notese que los dos ultimos se definen en funcion del primero por lo que una formalizacion matematica precisa del concepto de funcion Σ-efectivamente computable, resuelve el problema de modelizar en forma matematica estos a tres conceptos.

En esta materia daremos las tres formalizaciones matematicas mas clasicas del concepto de funcion Σ-efectivamente computable. La primera y la mas apegada a la idea intuitiva de procedimiento efectivo es la dada por Alan Turing via la matematizacion del concepto de maquina y es tema central de esta guia. La segunda, es la dada por Godel en su estudio de sistemas formales de la logica de primer orden. Por ultimo veremos la formalizacion via un lenguaje de programacion imperativo. En honor a la influencia que tuvo Von Neumann en el diseño de la primer computadora de caracter universal (i.e. programable de proposito general), llamaremos a este paradigma el paradigma imperativo de Von Neumann.

El paradigma de Turing

Estudiaremos el concepto de maquina de Turing, el cual fue introducido por Alam Turing para formalizar o modelizar matematicamente la idea de procedimiento efectivo. Una vez definidas las maquinas podremos dar una modelizacion matematica precisa del concepto de funcion Σ-efectivamente computable. Llamaremos a estas funciones Σ-Turing computables y seran aquellas que (en algun sentido que sera bien precisado matematicamente) pueden ser computadas por una maquina de Turing. Por supuesto, la fidedignidad de este concepto, es decir cuan buena es la modelizacion matematica dada por Turing, puede no ser clara al comienzo pero a medida que vayamos avanzando en nuestro estudio y conozcamos ademas los otros paradigmas y su relacion, quedara claro que el modelo de Turing es acertado.

Vivimos en un mundo plagado de maquinas (ascensores, celulares, relojes, taladros, etc). Una caracteristica comun a todas las maquinas es que tienen distintos estados posibles. Un estado es el conjunto de caracteristicas que determinan un momento concreto posible de la maquina cuando esta funcionando. Por ejemplo un estado posible de un ascensor seria:

- esta en el tercer piso, con la primer puerta abierta y la otra cerrada, esta apretado el boton de ir al sexto piso, etc

donde ponemos etc porque dependiendo del tipo de ascensor (si es con memoria, a que pisos puede ir, etc) habra mas datos que especificar para determinar un estado concreto.

Otra caracteristica comun de las maquinas es que interactuan de distintas formas con el usuario o mas generalmente su entorno. Dependiendo de que accion se ejecute sobre la maquina y en que estado este, la maquina realizara alguna tarea y ademas cambiara de estado. En general las maquinas son deterministicas en el sentido que siempre que esten en determinado estado y se les aplique determinada accion, realizaran la misma tarea y pasaran al mismo estado.

Descripcion informal de las maquinas de Turing

Son un modelo abstracto de maquina con una cantidad finita de estados la cual trabaja sobre una cinta de papel dividida en cuadros e interactua o recibe acciones externas por medio de una cabeza lectora que lee de a un cuadro de la cinta a la ves y ademas puede borrar el contenido del cuadro leido y escribir en el un simbolo. Tambien la cabeza lectora puede moverse un cuadro hacia la izquierda o hacia la derecha. La cinta tiene un primer cuadro hacia su izquierda pero hacia la derecha puede extenderse todo lo necesario. En un cuadro de la cinta podra haber un simbolo o un cuadro puede simplemente estar en blanco. Es decir que habra un alfabeto Γ el cual consiste de todos los simbolos que pueden figurar en la cinta. Esto sera parte de los datos o caracteristicas de cada maquina, es decir, Γ puede cambiar dependiendo de la maquina. La maquina, en funcion del estado en que se encuentre y de lo que vea su cabeza lectora en el cuadro escaneado, podra moverse a lo sumo un cuadro (izquierda, derecha o quedarse quieta), modificar lo que encuentre en dicho cuadro (borrando y escribiendo algun nuevo simbolo) y cambiar de estado (posiblemente al mismo que tenia). Para simplificar supondremos que hay en Γ un simbolo el cual si aparece en un cuadro de la cinta, significara que dicho cuadro esta sin escribir o en blanco. Esto nos permitira describir mas facilmente el funcionamiento de la maquina. En gral llamaremos B a tal simbolo. Tambien por lo general llamaremos Q al conjunto de estados de la maquina.

Tambien cada maquina tendra un estado especial el cual sera llamado su estado inicial, generalmente denotado con q_0, el cual sera el estado en el que estara la maquina al comenzar a trabajar sobre la cinta. Hay otras caracteristicas que tendran las maquinas de Turing pero para dar un primer ejemplo ya nos basta. Describiremos una maquina de Turing M que tendra $\Gamma = \{@, a, b, B\}$ y tendra dos estados, es decir $Q = \{q_0, q_1\}$. Obviamente q_0 sera su estado inicial y ademas su "comportamiento" estara dado por las siguientes clausulas:

- Estando en estado q_0 si ve ya sea b o B o @, entonces se queda en estado q_0 y se mueve a la derecha
- Estando en estado q_0 si ve a entonces reescribe @, se mueve a la izquierda y cambia al estado q_1
- Estando en estado q_1 si ve a o b o B o @ entonces lo deja como esta, se mueve a la izquierda y queda en estado q_1

Supongamos ahora que tomamos una palabra $\alpha \in \Gamma^*$ y la distribuimos en la cinta dejando el primer cuadro en blanco y luego poniendo los simbolos de α en los siguientes cuadros. Supongamos ademas que ponemos la maquina en estado q_0 y con su cabeza lectora escaneando el primer cuadro de la cinta. Esto lo podemos representar graficamente de la siguiente manera

$$\begin{array}{llllllll} B & \alpha_1 & ... & \alpha_n & B & B & B & ... \\ \uparrow \\ q_0 \end{array}$$

donde $\alpha_1, ..., \alpha_n$ son los sucesivos simbolos de α. Supongamos ademas que a ocurre an α. Dejamos al lector ir aplicando las clausulas de M para convencerse que luego de un rato de funcionar M, la situacion sera

$$\begin{array}{llllllll} B & \beta_1 & ... & \beta_n & B & B & B & ... \\ \uparrow \\ q_1 \end{array}$$

donde $\beta_1 ... \beta_n$ es el resultado de reemplazar en α la primer ocurrencia de a por @.

Ejercicio 1: Que sucede cuando a no ocurre en α?

Una cosa que puede pasar es que para un determinado estado p y un $\sigma \in \Gamma$, la maquina no tenga contemplada ninguna accion posible. Por ejemplo sea M la maquina de Turing dada por $Q = \{q_0\}$, $\Gamma = \{@, \$, B\}$ y por la siguiente clausula:

- Estando en estado q_0 si ve ya sea @ o B, entonces se queda en estado q_0 y se mueve a la derecha

Es facil ver que si partimos de una situacion

$$\begin{array}{lllllll} B & \alpha_1 & ... & \alpha_n & B & B & B & ... \\ \uparrow \\ q_0 \end{array}$$

donde $\alpha_1, ..., \alpha_n \in \Gamma$, entonces si ningun α_i es igual a \$, la maquina se movera indefinidamente hacia la derecha y en caso contrario se movera i pasos a la derecha y se detendra, donde i es el menor l tal que $\alpha_i = \$$.

Otro caso posible de detencion de una maquina de Turing es cuando esta escaneando el primer cuadro de la cinta y su unica accion posible implica moverse un cuadro a la izquierda. Tambien en estos casos diremos que la maquina se detiene ya que la cinta no es extensible hacia la izquierda.

Otra caracteristica de las maquinas de Turing es que poseen un *alfabeto de entrada* el cual esta contenido en el alfabeto Γ y en el cual estan los simbolos que se usaran para formar la configuracion inicial de la cinta (exepto B). En general lo denotaremos con Σ al alfabeto de entrada y los simbolos de $\Gamma - \Sigma$ son considerados auxiliares. Tambien habra un conjunto F contenido en el conjunto Q de los estados de la maquina, cuyos elementos seran llamados *estados finales.*

Diremos que una palabra $w = w_1...w_n \in \Sigma^*$ es *aceptada por* M (*por alcance de estado final*) si partiendo de

$$\begin{array}{llllllll} B & w_1 & ... & w_n & B & B & B & ... \\ \uparrow \\ q_0 \end{array}$$

en algun momento de la computacion M esta en un estado de F. Llamaremos $L(M)$ al conjunto formado por todas las palabras que son aceptadas por alcance de estado final

Ejercicio 2: Para cada uno de los siguientes lenguajes, encuentre una máquina de Turing M tal que $L(M)$ sea dicho lenguaje

(a) $\{w \in \{a,b\}^* : a \text{ ocurre en } w\}$

(b) $\{ab\}$

(c) $\{a^n b^n : n \geq 2\}$ (explicada en video en granlogico.com)

(d) $\{w \in \{a,b\}^* : |w|_a \text{ es par y } |w|_b \text{ es impar}\}$

Diremos que una palabra $w = w_1...w_n \in \Sigma^*$ es *aceptada por* M (*por detencion*) si partiendo de

$$\begin{array}{llllllll} B & w_1 & ... & w_n & B & B & B & ... \\ \uparrow \\ q_0 \end{array}$$

en algun momento M se detiene. Llamaremos $H(M)$ al conjunto formado por todas las palabras que son aceptadas por detencion

Ejercicio 3: Para cada uno de los lenguajes del ejercicio anterior encuentre una máquina de Turing M tal que $H(M)$ sea dicho lenguaje (hint: modifique adecuadamente cada una de las maquinas construidas para el ejercicio anterior)

Definicion matematica de maquina de Turing

Si bien las maquinas usadas por Turing para simular procedimientos efectivos son deterministicas, estudiaremos aqui tambien a las maquinas de Turing no deterministicas y sus lenguajes aceptados, dada su importancia en un tema central a las ciencias de la computacion como la complejidad.

Una *maquina de Turing* es una 7-upla $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ donde

- Q es un conjunto finito cuyos elementos son llamados *estados*
- Γ es un alfabeto que contiene a Σ
- Σ es un alfabeto llamado el *alfabeto de entrada*
- $B \in \Gamma - \Sigma$ es un simbolo de Γ llamado el *blank symbol*
- $\delta : Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R, K\})$
- q_0 es un estado llamado el *estado inicial* de M
- $F \subseteq Q$ es un conjunto de estados llamados *finales*

Si bien en nuestra definicion de maquina de Turing no hay ninguna restriccion acerca de la naturaleza de los elementos de Q, para continuar nuestro analisis asumiremos siempre que Q es un alfabeto disjunto con Γ. Esto nos permitira dar definiciones matematicas precisas que formalizaran el funcionamiento de las maquinas de Turing en el contexto de las funciones mixtas. Deberia quedar claro que el hecho que solo analicemos maquinas en las cuales Q es un alfabeto disjunto con Γ, no afectara la profundidad y generalidad de nuestros resultados.

Ejercicio 4: V o F o I, justifique.

(a) Si $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ es una maquina de Turing, entonces δ es una funcion $(Q \cup \Gamma)$-mixta

(b) Si $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ es una maquina de Turing, entonces D_δ es un conjunto $(Q \cup \Gamma)$-mixto

(c) Si $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ es una maquina de Turing, entonces I_δ es un conjunto $(Q \cup \Gamma)$-mixto

Maquinas de Turing deterministicas

Una maquina de Turing M sera llamada *deterministica* cuando se de que $|\delta(p,\sigma)| \leq 1$, cualesquiera sean $p \in Q$ y $\sigma \in \Gamma$.

Ejercicio 5: V o F o I, justifique

(a) Si $M = (Q,\Sigma,\Gamma,\delta,q_0,B,F)$ es una maquina de Turing deterministica, entonces δ es una funcion $(Q \cup \Gamma)$-mixta

Descripciones instantaneas

Una *descripcion instantanea* sera una palabra de la forma $\alpha q \beta$, donde $\alpha, \beta \in \Gamma^*$, $[\beta]_{|\beta|} \neq B$ y $q \in Q$. La descripcion instantanea $\alpha_1 ... \alpha_n q \beta_1 ... \beta_m$, con $\alpha_1, ..., \alpha_n$, $\beta_1, ..., \beta_m \in \Gamma$, $n, m \geq 0$ representara la siguiente situacion

$$\begin{array}{cccccccccccc} \alpha_1 & \alpha_2 & ... & \alpha_n & \beta_1 & \beta_2 & ... & \beta_m & B & B & B & ... \\ & & & & \uparrow & & & & & & & \\ & & & & q & & & & & & & \end{array}$$

Usaremos Des para denotar el conjunto de las descripciones instantaneas. Definamos la funcion $St : Des \rightarrow Q$, de la siguiente manera

$$St(d) = \text{unico simbolo de } Q \text{ que ocurre en } d$$

Ejercicio 6: V o F o I, justifique.

(a) Si d es una descripcion instantanea, entonces $Ti(d) = 3-$UPLA

(b) Si d es una descripcion instantanea, entonces $St(d) = d \cap Q$

La relacion $\vdash$

Dado $\alpha \in (Q \cup \Gamma)^*$, definamos $\lfloor \alpha \rfloor$ de la siguiente manera

$$\begin{array}{rcl} \lfloor \varepsilon \rfloor & = & \varepsilon \\ \lfloor \alpha\sigma \rfloor & = & \alpha\sigma, \text{ si } \sigma \neq B \\ \lfloor \alpha B \rfloor & = & \lfloor \alpha \rfloor \end{array}$$

Es decir $\lfloor \alpha \rfloor$ es el resultado de remover de α el tramo final mas grande de la forma B^n. Dada cualquier palabra α definimos

$$^{\curvearrowright}\alpha = \begin{cases} [\alpha]_2 ... [\alpha]_{|\alpha|} & \text{si } |\alpha| \geq 2 \\ \varepsilon & \text{si } |\alpha| \leq 1 \end{cases} \qquad \alpha^{\curvearrowleft} = \begin{cases} [\alpha]_1 ... [\alpha]_{|\alpha|-1} & \text{si } |\alpha| \geq 2 \\ \varepsilon & \text{si } |\alpha| \leq 1 \end{cases}$$

Dadas $d_1, d_2 \in Des$, escribiremos $d_1 \vdash d_2$ cuando existan $\sigma \in \Gamma$, $\alpha, \beta \in \Gamma^*$ y $p, q \in Q$ tales que se cumple alguno de los siguientes casos

Caso 1.

$$\begin{array}{rcl} d_1 & = & \alpha p \beta \\ (q, \sigma, R) & \in & \delta\left(p, [\beta B]_1\right) \\ d_2 & = & \alpha \sigma q \frown \beta \end{array}$$

Caso 2.

$$\begin{array}{rcl} d_1 & = & \alpha p \beta \\ (q, \sigma, L) & \in & \delta\left(p, [\beta B]_1\right) \text{ y } \alpha \neq \varepsilon \\ d_2 & = & \left\lfloor \alpha \frown q\, [\alpha]_{|\alpha|}\, \sigma \frown \beta \right\rfloor \end{array}$$

Caso 3.

$$\begin{array}{rcl} d_1 & = & \alpha p \beta \\ (q, \sigma, K) & \in & \delta(p, [\beta B]_1) \\ d_2 & = & \lfloor \alpha q \sigma \frown \beta \rfloor \end{array}$$

Escribiremos $d \nvdash d'$ para expresar que no se da $d \vdash d'$. Para $d, d' \in Des$ y $n \geq 0$, escribiremos $d \overset{n}{\vdash} d'$ si existen $d_1, ..., d_{n+1} \in Des$ tales que

$$\begin{array}{rcl} d & = & d_1 \\ d' & = & d_{n+1} \\ d_i & \vdash & d_{i+1}, \text{ para } i = 1, ..., n. \end{array}$$

Notese que $d \overset{0}{\vdash} d'$ sii $d = d'$. Finalmente definamos

$$d \overset{*}{\vdash} d' \text{ sii } (\exists n \in \omega)\ d \overset{n}{\vdash} d'.$$

Ejercicio 7: V o F o I, justifique.

(a) $d \vdash d$, para cada $d \in Des$

(b) Si $\alpha p \beta \nvdash d$ para toda descripción instantánea d entonces $\delta(p, [\beta B]_1) = \emptyset$.

(c) Si $(p, a, L) \in \delta(p, a)$ entonces $pa \nvdash d$ para toda descripción instantánea d.

(d) Dadas $d, d' \in Des$, se tiene que si $d \vdash d'$, entonces $|d| \leq |d'| + 1$

Ejercicio 8: Pruebe que M es deterministica sii para cada $d \in Des$ hay a lo sumo una $d' \in Des$ tal que $d \vdash d'$

Detencion

Dada $d \in Des$, diremos que M *se detiene partiendo de* d si existe $d' \in Des$ tal que

- $d \overset{*}{\vdash} d'$
- $d' \nvdash d''$, para cada $d'' \in Des$

Ejercicio 9: Estudie los dos posibles casos de detencion:

(a) estando el cabezal sobre el primer cuadro de la cinta

(b) estando el cabezal en un cuadro que no es el primero

Ejercicio 10: V o F o I, justifique.

(a) Sea $d \in Des$. Entonces si existe una infinitupla $(d_1, d_2, ...)$ tal que $d \vdash d_1 \vdash d_2 \vdash d_3 \vdash d_4 \vdash \cdots$, se tiene que M no se detiene partiendo de d

(b) Supongamos que para cada $(p, \sigma) \in Q \times \Gamma$ la tercera coordenada de cada elemento de $\delta(p, \sigma)$ es igual a L. Entonces M se detiene partiendo de cada $d \in Des$

(c) Si $d = \alpha p \beta$ es una descripcion instantamea tal que $d \nvdash d'$, para cada descripcion instantanea d', entonces $\delta(p, [\beta B]_1) = \emptyset$

El lenguaje $L(M)$

Diremos que una palabra $w \in \Sigma^*$ es *aceptada por* M (*por alcance de estado final*) cuando

$$\lfloor q_0 B w \rfloor \overset{*}{\vdash} d, \text{ con } d \text{ tal que } St(d) \in F.$$

El *lenguage aceptado por* M (*por alcance de estado final*) se define de la siguiente manera

$$L(M) = \{w \in \Sigma^* : w \text{ es aceptada por } M \text{ por alcance de estado final}\}.$$

Ejercicio 11: Para cada uno de los siguientes lenguajes, encuentre una máquina de Turing M tal que $L(M)$ sea dicho lenguaje

(a) $\{w \in \{a, b\}^* : |w|_a = 2\,|w|_b\}$

(b) $\left\{a^i b^j c^k : i \neq j \text{ o } j \neq k\right\}$

(c) $\left\{w \in \{a,b\}^* : w = w^R\right\}$ (palabras capicuas)

Ejercicio 12: V o F o I, justifique.

(a) Si q_2 es un estado final de la máquina M, $(q_1, B, R) \in \delta(q_0, B)$ y $(q_2, b, R) \in \delta(q_1, a)$ entonces $a \in L(M)$.

(b) Si q_2 es un estado final de la máquina M, $(q_1, B, R) \in \delta(q_0, B)$ y $(q_2, b, R) \in \delta(q_1, a)$ entonces $b \in L(M)$.

(c) Supongamos existe una infinitupla $(d_1, d_2, ...)$ tal que

i. $St(d_i) \notin F$, para cada $i = 1, 2, ...$

ii. $\lfloor q_0 Bw \rfloor \vdash d_1 \vdash d_2 \vdash d_3 \vdash d_4 \vdash \cdots$

Entonces se tiene que $w \notin L(M)$

El lenguaje $H(M)$

Diremos que una palabra $w \in \Sigma^*$ es *aceptada por M (por detencion)* cuando M se detiene partiendo de $\lfloor q_0 Bw \rfloor$. El *lenguage aceptado por M (por detencion)* se define de la siguiente manera

$$H(M) = \{w \in \Sigma^* : w \text{ es aceptada por } M \text{ por detencion}\}$$

Ejercicio 13: Para cada uno de los lenguajes del Ejercicio 11 encuentre una máquina de Turing M tal que $H(M)$ sea dicho lenguaje (hint: modifique adecuadamente cada una de las maquinas construidas para el Ejercicio 11)

Ejercicio 14: V o F o I, justifique.

(a) Si q_2 es un estado final de la máquina M, $(q_1, B, R) \in \delta(q_0, B)$ y $(q_2, b, R) \in \delta(q_1, a)$ entonces $a \in H(M)$.

(b) Si q_2 es un estado final de la máquina M, $(q_1, B, R) \in \delta(q_0, B)$ y $(q_2, b, K) \in \delta(q_1, a)$ entonces $a \in H(M)$.

(c) Sea $w \in \Sigma^*$. Entonces si existe una infinitupla $(d_1, d_2, ...)$ tal que $\lfloor q_0 Bw \rfloor \vdash d_1 \vdash d_2 \vdash d_3 \vdash d_4 \vdash \cdots$,se tiene que $w \notin H(M)$

Aceptaremos sin demostracion el siguiente resultado.

Lemma 1 *Sea $L \subseteq \Sigma^*$. Entonces son equivalentes*

(1) Existe una maquina de Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ tal que $L = L(M)$

(2) Existe una maquina de Turing $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ tal que $L = H(M)$

Funciones Σ-Turing computables

Para poder computar funciones mixtas con una maquina de Turing necesitaremos un simbolo para representar numeros sobre la cinta. Llamaremos a este simbolo *unit* y lo denotaremos con $\shortmid$. Mas formalmente una *maquina de Turing con unit* es una 8-upla $M = (Q, \Sigma, \Gamma, \delta, q_0, B, \shortmid, F)$ tal que $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$ es una maquina de Turing y $\shortmid$ es un simbolo distingido perteneciente a $\Gamma - (\{B\} \cup \Sigma)$.

Diremos que una funcion $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to \Sigma^*$ es Σ-*Turing computable* si existe una maquina de Turing deterministica con unit, $M = (Q, \Sigma, \Gamma, \delta, q_0, B, \shortmid, F)$ tal que:

(1) Si $(\vec{x}, \vec{\alpha}) \in D_f$, entonces hay un $p \in Q$ tal que

$$\lfloor q_0 B \shortmid^{x_1} B ... B \shortmid^{x_n} B\alpha_1 B ... B\alpha_m \rfloor \overset{*}{\vdash} \lfloor pBf(\vec{x}, \vec{\alpha}) \rfloor$$

y $\lfloor pBf(\vec{x}, \vec{\alpha}) \rfloor \not\vdash d$, para cada $d \in Des$

(2) Si $(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m} - D_f$, entonces M no se detiene partiendo de

$$\lfloor q_0 B \shortmid^{x_1} B ... B \shortmid^{x_n} B\alpha_1 B ... B\alpha_m \rfloor .$$

En forma similar, una funcion $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to \omega$, es llamada Σ-*Turing computable* si existe una maquina de Turing deterministica con unit, $M = (Q, \Sigma, \Gamma, \delta, q_0, B, \shortmid, F)$, tal que:

(1) Si $(\vec{x}, \vec{\alpha}) \in D_f$, entonces hay un $p \in Q$ tal que

$$\lfloor q_0 B \shortmid^{x_1} B ... B \shortmid^{x_n} B\alpha_1 B ... B\alpha_m \rfloor \overset{*}{\vdash} \left\lfloor pB \shortmid^{f(\vec{x}, \vec{\alpha})} \right\rfloor$$

y $\left\lfloor pB \shortmid^{f(\vec{x}, \vec{\alpha})} \right\rfloor \not\vdash d$, para cada $d \in Des$

(2) Si $(\vec{x}, \vec{\alpha}) \in \omega^n \times \Sigma^{*m} - D_f$, entonces M no se detiene partiendo de

$$\lfloor q_0 B \shortmid^{x_1} B ... B \shortmid^{x_n} B\alpha_1 B ... B\alpha_m \rfloor$$

Cuando M y f cumplan los items (1) y (2) de la definicion anterior, diremos que la funcion f es *computada* por M.

Por supuesto esta definicion no tendria sentido como modelo matematico del concepto de funcion Σ-efectivamente computable si no sucediera que toda funcion Σ-Turing computable fuera Σ-efectivamente computable. Este hecho es intuitivamente claro y lo expresamos en forma de proposicion.

Proposition 2 *Si $f : D_f \subseteq \omega^n \times \Sigma^{*m} \to O$, con $O \in \{\omega, \Sigma^*\}$, es computada por una maquina de Turing deterministica con unit $M = (Q, \Sigma, \Gamma, \delta, q_0, B, \shortmid, F)$, entonces f es Σ-efectivamente computable.*

Proof. Haremos el caso $O = \Sigma^*$. Sea $\mathbb{P}$ el siguiente procedimiento efectivo.

- Conjunto de datos de entrada de $\mathbb{P}$ igual a $\omega^n \times \Sigma^{*m}$
- Conjunto de datos de salida de $\mathbb{P}$ contenido en O
- Funcionamiento: Hacer funcionar paso a paso la maquina M partiendo de la descripcion instantanea $\lfloor q_0 B \shortmid^{x_1} B ... B \shortmid^{x_n} B\alpha_1 B ... B\alpha_m \rfloor$. Si en alguna instancia M termina, dar como salida el resultado de remover de la descripcion instantanea final los dos primeros simbolos.

Notese que este procedimiento termina solo en aquelos elementos $(\vec{x}, \vec{\sigma}) \in \omega^n \times \Sigma^{*m}$ tales que la maquina M termina partiendo desde

$$\lfloor q_0 B \shortmid^{x_1} B ... B \shortmid^{x_n} B\alpha_1 B ... B\alpha_m \rfloor$$

por lo cual termina solo en los elementos de D_f ya que M computa a f. Ademas es claro que en caso de terminacion el procedimiento da como salida $f(\vec{x}, \vec{\sigma})$. $\blacksquare$

Sin envargo el modelo Turingniano podria a priori no ser del todo correcto ya que podria pasar que haya una funcion que sea computada por un procedimiento efectivo pero que no exista una maquina de Turing que la compute. En otras palabras el modelo Turingniano podria ser incompleto. La completitud de este modelo puede no ser clara al comienzo pero a medida que vayamos avanzando en nuestro estudio y conozcamos ademas los otros paradigmas y su relacion, quedara claro que el modelo de Turing es acertado.

Ejercicio 15: Sea $\Sigma = \{a, b\}$. Para cada una de las siguientes funciones Σ-mixtas dar una máquina de Turing $(Q, \Gamma, \Sigma, \delta, q_0, B, \shortmid, \emptyset)$ que la compute

(a) $\begin{array}{rcl} Suc : \omega & \rightarrow & \omega \\ n & \rightarrow & n+1 \end{array}$

(b) $\begin{array}{rcl} Pred : \mathbf{N} & \rightarrow & \omega \\ n & \rightarrow & n-1 \end{array}$

(c) $\begin{array}{rcl} p_2^{1,1} : \omega \times \Sigma^* & \rightarrow & \Sigma^* \\ (x, \alpha) & \rightarrow & \alpha \end{array}$

(explicada en video en granlogico.com)

(d) $\begin{array}{rcl} f : D_f & \rightarrow & \omega \\ (x, \alpha, \beta) & \rightarrow & x + |\alpha| + |\beta| \end{array}$

donde $D_f = \{(x, \alpha, \beta) \in \omega \times \Sigma^{*2} : |\alpha| \text{ es impar}\}$

(e) $\begin{array}{rcl} C_2^{1,1} : \omega \times \Sigma^* & \rightarrow & \omega \\ (x, \alpha) & \rightarrow & 2 \end{array}$

(f) $\begin{array}{rcl} R : \{w \in \Sigma^+ : \exists x \in \Sigma^* \text{ tal que } w = xx\} & \rightarrow & \Sigma^* \\ \alpha & \rightarrow & \text{único } x \text{ tal que } w = xx \end{array}$

Ejercicio 16: V o F o I, justifique.

(a) Sea $M = (Q, \Sigma, \Gamma, \delta, q_0, B, \shortmid, F)$ una maquina de Turing deterministica con unit y supongamos que M computa a f. Entonces $D_f = \{d \in Des : M$ se detiene partiendo desde $d\}$

(b) Si f y g son dos funciones y M es es una máquina de Turing que computa a f y a g entonces $f = g$.

(c) Sea $M = (Q, \Sigma, \Gamma, \delta, q_0, B, \shortmid, F)$ una maquina de Turing deterministica con unit y supongamos que M computa a f y que f es Σ-total. Entonces M se detiene partiendo desde d, cualesquiera sea $d \in Des$

Ejercicio 17: Como se vio anteriormente el modelo de Turing del concepto de funcion Σ-efectivamente computable es el concepto matematico de funcion Σ-Turing computable. Cual seria el modelo de Turing del concepto de conjunto Σ-efectivamente computable? que nombre le pondria?

Ejercicio 18: Idem que el ejercicio anterior pero para "enumerable"